(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

特開平9－205422

(43)公開日　平成９年(1997)８月５日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｈ０４Ｌ	9/30			Ｈ０４Ｌ	9/00	６６３Ｚ
Ｇ０９Ｃ	1/00	６２０	7259－5Ｊ	Ｇ０９Ｃ	1/00	６２０Ｚ
		６３０	7259－5Ｊ			６３０Ｄ
			7259－5Ｊ			６３０Ｆ
Ｈ０４Ｌ	9/08			Ｈ０４Ｌ	9/00	６０１Ｄ

審査請求　未請求　請求項の数18　ＯＬ　（全 24 頁）　最終頁に続く

(21)出願番号　特願平8－5277

(22)出願日　平成８年(1996)１月16日

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレイション
INTERNATIONAL BUSINESS MASCHINES CORPORATION
アメリカ合衆国10504、ニューヨーク州　アーモンク　（番地なし）

(72)発明者　アミール・ヘルツベルグ
アメリカ合衆国ニューヨーク州、ブロンクス、ブラックストーン・アベニュー　3935

(74)代理人　弁理士　合田　潔　（外２名）

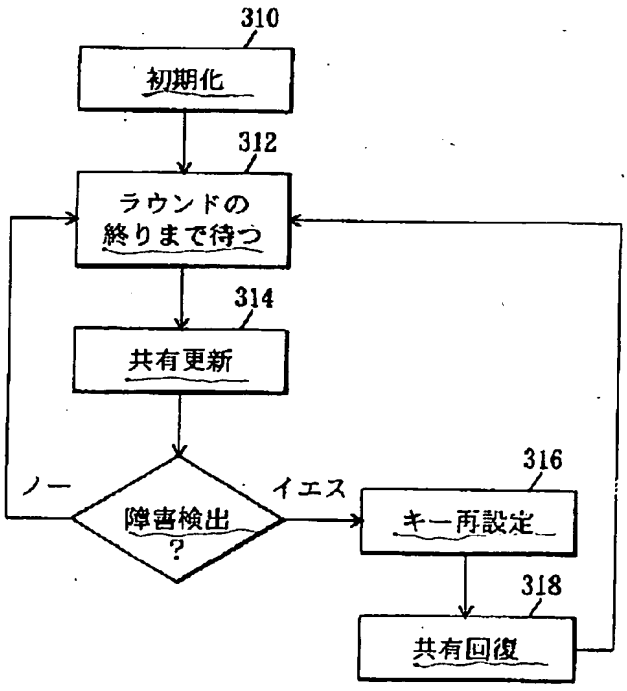
最終頁に続く

(54)【発明の名称】　順向性、ロバスト及び回復可能な分散しきい値秘密共有を有する公開キー暗号システム及びその方法

(57)【要約】

【課題】　順向性、ロバスト及び回復可能な分散しきい値秘密共有方式を有する公開キー暗号システム及びその方法の提供。

【解決手段】順向性しきい値秘密共有暗号システムはサーバーのセットを用いる。ｎサーバーのうちの少なくとも(k+1）のサーバーが活動状態であり且つ正直である場合にサービスが維持される。秘密署名キーは、敵対者が少なくとも(k+1）サーバーに侵入した場合にのみコンプロマイズされる。また、正直なサーバーが誤りのあるサーバーを検出し、サービスは中断されない。さらに、敵対者がコンプロマイズしたサーバーにある全ての局所情報を敵対者が消去した場合でも、サーバーは正しいプロトコルの実行に復帰すると直ちに前記情報を復元できる。本発明の方法及びシステムは順向性を有するため、敵対者が秘密を知るためには、アルゴリズムの同じラウンドの間に(k+1）サーバーへの侵入が必要となる。



【特許請求の範囲】
【請求項１】順向性、ロバスト及び回復可能な分散しきい値秘密共有を有する公開キー暗号化の方法であって、
キーを形成するために通信ネットワークによりリンクされたサーバーを初期化するステップと、
終了を有する別個のラウンドで動作するように前記サーバーを同期させるステップと、
前記通信ネットワークで同報通信されたメッセージから前記ラウンドの前記終了で更新されたキーを計算するステップと、
コンプロマイズされたサーバーのセットを形成するために前記更新されたキーを検査するステップと、
前記コンプロマイズされたサーバーのセットを回復するステップとを含む方法。
【請求項２】前記初期化するステップは、
前記サーバーの各々の乱数を選択するステップと、
前記乱数から前記サーバーの各々の秘密値を計算するステップと、
前記秘密値から前記サーバーの各々の秘密キーを計算するステップと、
前記秘密キーの公開対応部を前記通信ネットワークで同報通信するステップとを含む、請求項１に記載の方法。
【請求項３】前記更新されたキーを計算するステップは、
前記サーバーの各々の多項式を定義する乱数のセットを取出すステップと、
前記多項式から導き出された、前記サーバーの各々の新しい秘密キーを取出すステップと、
前記多項式から導き出されたメッセージを前記通信ネットワークで同報通信するステップとを含む、請求項１に記載の方法。
【請求項４】前記更新されたキーを検査する前記ステップは、
無効なサーバーのセットを形成するために前記通信ネットワークで同報通信された前記メッセージを分析するステップと、
前記無効なサーバーのセットに対応する告発のセットを生成するステップと、
前記告発のセットを前記通信ネットワークで同報通信するステップと、
更新された無効なサーバーのセットを形成する前記同報通信された告発のセットを分析するステップとを含む、請求項１に記載の方法。
【請求項５】前記コンプロマイズされたサーバーのセットを識別するメッセージをコンソールに表示するステップを更に含む、請求項１に記載の方法。
【請求項６】前記サーバーのセットを回復するステップは、
前記コンプロマイズされたサーバーのセットにある各サーバーに新しい秘密キーを設定するステップと、
前記サーバーから回復サーバーのセットを選択するステップと、
前記回復サーバーのセットにある各サーバーの下位共有を計算するステップと、
前記下位共有から取出されたメッセージを前記通信ネットワークで同報通信するステップと、
前記コンプロマイズされたサーバーのセットが受取った前記下位共有から取出された前記メッセージを検査するステップとを含む、請求項１に記載の方法。
【請求項７】回復サーバーのセットを選択する前記ステップは、
前記サーバーの下位セットを選択するステップと、
前記下位セットの各サーバーの乱数値のセットを計算するをステップと、
前記乱数値のセットから取出された署名されたメッセージを前記通信ネットワークで同報通信するステップと、
無効サーバーのセットを取出すために前記署名されたメッセージを検査するステップと、
前記サーバーの前記下位セットから前記無効サーバーのセットを除去するステップとを含む、請求項６に記載の方法。
【請求項８】前記サーバーの各々の乱数を取出すステップと、
前記乱数から前記サーバー毎の第１の署名の部分を計算するステップと、
前記第１の署名の部分を前記通信ネットワークで同報通信するステップと、
前記第１の署名の部分から前記サーバーの各々の第２の署名の部分を計算するステップとを更に含む、請求項１に記載の方法。
【請求項９】署名されたメッセージを前記第２の署名の部分により検査するステップを更に含む、請求項８に記載の方法。

システム・クレーム
【請求項１０】順向性、ロバスト及び回復可能な分散しきい値秘密共有を有する公開キー暗号方式を処理するデータ処理システムであって、
通信ネットワークによりリンクされたサーバーと、
前記サーバーと関連したキーを形成するために前記サーバーを初期化する初期化手段と、
終了を有する別個のラウンドに前記サーバーの動作を同期させるタイミング手段と、
更新されたキーを生成するために前記別個のラウンドの各ラウンドの終了で前記キーを更新する更新手段と、
コンプロマイズされたサーバーのセットを形成するために前記更新されたキーを検査する検査手段と、
前記コンプロマイズされたサーバーのセットを回復する回復手段とを備えるデータ処理システム。
【請求項１１】前記サーバーの各々の乱数を選択する選択手段と、
前記乱数から前記サーバーの各々の秘密値を計算する第

１の計算手段と、
前記秘密値から前記サーバーの各々の秘密キー及び前記秘密キーの公開対応部を計算する第２の計算手段と、
前記秘密キーの各々の前記公開対応部を前記通信ネットワークで同報通信する同報通信手段とを備える、請求項１０に記載のデータ処理システム。
【請求項１２】前記更新手段は、
前記サーバーの各々に関連した、多項式を定義する乱数のセットを生成する乱数生成手段と、
前記多項式から前記サーバーの各々の新しい秘密キーを生成する秘密キー生成手段と、
前記通信ネットワークで前記多項式から取出されたメッセージを同報通信する同報通信手段とを備える、請求項１０に記載のデータ処理システム。
【請求項１３】前記検査手段は、
無効なサーバーのセットを形成するために前記通信ネットワークで同報通信されたメッセージを分析する第１の分析手段と、
前記無効なサーバーのセットに対応する告発のセットを生成する告発手段と、
前記告発のセットを前記通信ネットワークで同報通信する同報通信手段と、
無効なサーバーの更新されたセットを形成するために前記同報通信された告発のセットを分析する第２の分析手段とを備える、請求項１０に記載のデータ処理システム。
【請求項１４】前記コンプロマイズされたサーバーのセットを表示する表示手段を更に備える、請求項１０に記載のデータ処理システム。
【請求項１５】前記回復手段は、
前記サーバーから回復サーバーのセットを選択する選択手段と、
前記回復サーバーのセットにあるサーバーの各々の下位共有を計算する計算手段と、
前記回復サーバーのセットにあるサーバーの各々の前記下位共有から取出されたメッセージを前記通信ネットワークで同報通信する同報通信手段と、
前記コンプロマイズされたサーバーのセットが受取った前記下位共有の各々から取出された前記メッセージを検査する検査手段とを更に備える、請求項１０に記載のデータ処理システム。
【請求項１６】前記選択手段は、
前記サーバーの下位セットを選択する選択手段と、
前記下位セットにある各サーバーの乱数値のセットを計算する計算手段と、
前記乱数値のセットから取出された署名されたメッセージを前記通信ネットワークで同報通信する同報通信手段と、
無効なサーバーのセットを取出すために前記署名されたメッセージを検査する検査手段と、
前記無効なサーバーのセットを前記下位セットから除去する除去手段とを備える、請求項１５に記載のデータ処理システム。
【請求項１７】前記サーバーの各々の乱数のセットを生成する乱数生成手段と、
前記乱数のセットから前記サーバーの各々の署名の第１の部分を計算する第１の計算手段と、
前記署名の第１の部分を前記通信ネットワークで同報通信する同報通信手段と、
前記第１の署名の部分から前記サーバーの各々の署名の第２の部分を計算する第２の計算手段とを更に備える、請求項１０に記載のデータ処理システム。
【請求項１８】署名されたメッセージを前記第２の署名の部分により検査する検査手段を更に備える、請求項１７に記載データ処理手段。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明はデータ処理システムに、より詳しくは情報保全のための暗号機構を備えるデータ処理システムに関する。
【０００２】
【従来の技術】暗号化されたメッセージを送受するために公開キー暗号システムが用いられる。公開キー暗号システムは、公開キーと呼ばれる公に入手できる値を用いる数理的なアルゴリズムの実行によりメッセージが暗号化されるシステムである。そして、受取人は秘密キーと呼ばれる専用の値を用いるアルゴリズムを実行することによりメッセージを解読する。公開キー暗号は暗号アルゴリズムＥ及び解読アルゴリズムＤの選択に依存し、Ｅが完全に解読されても、Ｄの取出しは実効的に不可能であるようにする。公開キー暗号の３つの必要条件は:
(1) Ｄ(Ｅ(Ｐ)) = Ｐ, ここで、Ｐはメッセージである;
(2) ＥからＤを演繹することは極めて困難である; 及び
(3) Ｐを攻撃してもＥを破壊できない
ことである。従って、公開キーは自由に配布できるが、秘密キーは、それを用いるエンティティにより秘密が保持されなければならない。侵入者がエンティティのメモリ内容をアクセスした場合、システム保全は侵害される。これは最初の公開キー特徴の全てにあてはまる。キー第三者委託システム及び”ミカリ(Micali)”の規則にかなった暗号システムでは、秘密キーは多くの部分に分割され、そして各部分は異なるエンティティにより保持される。しかしながら、時間の経過とともに侵入者が各エンティティのメモリを読取りできる場合、システム保全は侵害される。
【０００３】［順向性秘密共有］順向性秘密共有の概念を説明するために、秘密共有方式の用語及びそれらの保全特性を説明する。
【０００４】［秘密共有におけるしきい値、ロバスト

(堅固性) 及び回復可能性] 秘密共有は最初に G.R. Blakley, Safeguarding Cryptographic Keys, AFIPSCon. Proc. (v.48), 1979, pp. 313-317及び A. Shamir, How to Share a Secret, Commun. ACM, 22, 1979, pp. 612-613 により導入された。その最も基本的な形式では、秘密共有は、ｎ人の参加者の間で秘密情報Ｍを分割し、参加者らが互いにそれを再構築できるようにする方法である。しかしながら、(n-1) の共有保持者のグループはどれもＭに関して何も知ることはできない。この機構は個々の情報を守る必要があるときは保全を高めるために必ず用いられる。保全を高めることは、サーバーの全てではないが、その幾つかに侵入できる敵対者から秘密を守りうることに相当する。あらゆる秘密共有方式は概念的に以下の段階を有する。

(1) 秘密を知っているディーラーがその共有を生成し、それらを共有保持者に配布するディーリング段階;
(2) 共有保持者の間で共有が維持される記憶段階;及び
(3) 共有保持者がかれらの共有から秘密を再構築する再構築段階

上記の基本的な秘密共有はあまり実際的ではない。なぜなら、敵対者が共有保持者をコンプロマイズし、当該保持者が保持した秘密共有を消去又は変更する場合、秘密Ｍは決して再構築できないからである。よって、上記の方式は、(n-1) サーバーに侵入してそれらの共有を知りうるが、サーバーのメモリ即ち記憶を消去又は変更することも、サーバーにクラッシュさせることもできず、そして共有の処理にも秘密の再構築にも干渉しない敵対者に対して安全である。

【０００５】基本的な方式の保全性を高める秘密共有の特性が定義される前に、サーバーに対する敵対者の異なるタイプの攻撃が定義されなければならない:

・敵対者がサーバーに侵入し、そのサーバーに記憶された全ての秘密情報: その秘密共有及び通信に用いるキーを知る場合、サーバーは敵対者にコンプロマイズされる。

・敵対者がサーバーに仕事を止めさせる場合、敵対者はサーバーを凍結する。しかしながら、敵対者が追放されると直ちに、サーバーは正しいプロトコルの実行に復帰できる（即ち、データは失われない）と想定される。サーバーのネットワークへのアクセスの中断はサーバーの凍結の例である。全ての所要のデータ（変数及びアルゴリズムコード）が消去又は変更されない場合でも、電力の遮断又はサーバー上の全プロセスの停止は凍結を構成する。凍結されたサーバーはメッセージを送受信せず、人の介入により再始動されるまで遊休状態である。

・サーバーが実行するプロトコルを敵対者が変更し、最初のプロトコルに関して間違っているメッセージをサーバーが送るようにする場合、敵対者はサーバーを制御する。他のサーバーの観点から、そのようなサーバーは不正をしている、即ち不正直である。凍結は制御の１つのケースである。

・サーバーが記憶している秘密データを敵対者がうまく消去又は変更したとき敵対者はサーバーを作動禁止する。プロトコルコードの消去又は他のサーバーの公開キーの消去は、システム管理がこのサーバーの活動を復帰させるのを困難にするが、この情報は公開されているので、システムオペレータに秘密情報を暴露せずに、それを再びサーバーに導入することができる。凍結及び作動禁止の間に質的な差異を作ることはサーバーの秘密共有の破壊である。

【０００６】上記のタイプの敵対者に対する保全を提供する、秘密共有方式の望ましい特性について以下に説明する。重要なことは、上記の基本的な秘密共有方式でのディーリング段階が保全されない状態における記憶段階及び再構築段階での保全である。この重要性は、ディーリング段階での保全を高めることを目的とする検証可能な秘密共有方式と対照的である。

【０００７】ｎサーバーのうちの(k+1) だけが秘密の再構築のために協同する必要がある場合、秘密共有方式はしきい値方式と呼ばれる。この特性は下記のように行動する攻撃者に対してシステムを保全する:

・再構築段階中にｎサーバーのうちの最大ｋサーバーまで凍結即ち作動禁止する。記憶段階中に、これらのサーバーが再構築段階開始前に提訴されることを条件に、敵対者はより多くのサーバーを凍結でき、しきい値方式は安全である。

・アルゴリズムの有効期間中、ｎサーバーのうちのｋサーバーをコンプロマイズする。ｎサーバーのうちの(k+1) サーバーをコンプロマイズすることは、敵対者が敵対者自身で秘密を再構築することを可能にする。上記の２つの必要条件は、しきい値ｋが厳密に少数である、即ち次の式が成立つ場合にのみ満たすことができる。

【数１】$2k + 1 \leqq n$

【０００８】最大ｋの不正なサーバーが存在しても再構築段階が安全である場合、秘密共有方式はロバストと呼ばれる。それ自身によるしきい値方式は、攻撃者が作動禁止し又は凍結しうるがコンプロマイズされたサーバーを制御できないとき、秘密を再構築するために障害のない(k+1) サーバーのグループを取出すのに問題がないことによる。なぜなら、障害は非活動状態を意味するからである。要するに、再構築段階で、全部でｋ以下のサーバーを作動禁止又は制御できる敵対者に対して、ロバストはしきい値方式を安全にする。

【０００９】しきい値秘密共有方式は、それが正しい秘密共有を失ったサーバーに復元できる場合に回復可能である。この方式は、正しい秘密共有を、それを失ったサーバーに復元できると仮定する。従って、回復機構は記憶段階で実行される。秘密共有の回復の特徴が本発明の目的である。この特徴は、記憶段階の時点のどこかでｎサーバーのうちのｋ以下のサーバーを作動禁止又は制御

できる敵対者に対してロバストしきい値方式を安全にする。あるサーバーが作動禁止されることをシステムが知る毎に回復段階は開始し、それがあまりに短いので、この段階中に敵対者は他のサーバーに移りえないと想定すれば、それは成功である。

【００１０】回復は秘密共有が失われる毎に必要であるが、それはこの秘密共有を知る敵対者がいてもいなくても起こりうる。コンプロマイズ及び作動禁止能力を有する敵対者がサーバーに侵入し、その秘密共有を知り、そしてそれを消去又は変更したとき、この共有は失われる。しかし、この共有は電力遮断によっても失われる。最初のケースで、敵対者が秘密を知った直後に、その秘密が公に回復されうることが論議されることがある。しかしながら、二人以上の敵対者がおり、公開な共有回復により、仕事の一部が他の敵対者に与えられることがある。また、電力遮断による作動禁止のケースを処理する機構が存在し、敵対者の一人が秘密を知った場合にその機構が用いられることがある。保全形式がどのケースを処理しているかを知ることは困難であるから、秘密共有が失われる毎に１つの保全回復機構を用いる方がよい。

【００１１】［秘密回復及びサーバー確認］敵対者が局所記憶装置及びコンプロマイズされたサーバーのメモリを消去又は変更する能力を有し、更に敵対者がサーバー間の通信チャンネルにメッセージを挿入しうる場合、完全な自動回復は不可能である。サーバーをリブートし、秘密共有アルゴリズムを実行するプロセスを開始し、このサーバー及び他のサーバーの間で相互確認する一定の手段を再び設けなければならない。

【００１２】秘密共有に関してサーバー確認の問題が議論されるのは稀である。しかしながら、サーバー確認は、秘密共有に参加するサーバーのグループの外部からサーバーにメッセージを送りうる敵対者が存在するとき、回復プロトコルの保全を保つために必要である。サーバーＡの回復は、その共有を再び設けようとするサーバーＢを確認する一定の手段を持たなければならない。また、サーバーＢは、それらが適切な共有を与えようとするサーバーが実際にサーバーＡであることを確認しなければならない。サーバーＡとサーバーＢとの間の安全なリンクにより、又はＡのサーバー及びＢのサーバーの間の秘密／公開の暗号解読及び署名キーの対により、これらの目標を達成することができる。しかしながら、サーバーＡがその秘密共有を失った場合、他のサーバーＢとのリンクの保全及び確認に用いる全ての他の秘密キーも失うことがある。これは、作動禁止されたサーバーに確認の手段を設ける際の人の介入が回復のために必要であることを意味する。この介入は、サーバーＡとサーバーＢとの間に新しいサーバー間リンク保全を設けること、又はサーバーＢの公開確認／暗号キーをサーバーＡに再び設ける際にその新しい秘密キーをサーバーＡに計算させてそれらを安全にサーバーＢに設けることを含む。順向システムの例で、秘密／公開キーがサーバー間通信に用いられるので、回復機構への人の介入は次のタイプである。

【００１３】［順向性］新しい秘密共有方式の特性は下記のように定義される。

順向性：順向性（プロアクティブ）は、敵対者がサーバーをコンプロマイズできる速度を制限することにより保全性を高める。順向秘密共有方式の記憶段階は、短い更新段階により分割されたラウンドからなる。この方式は、同じラウンドでｋ以下のサーバーをコンプロマイズできる敵対者に対して安全である。順向性なしに、この方式は、全記憶段階、即ち、実効的にアルゴリズムの有効期間中、最大ｋサーバーをコンプロマイズできる敵対者に対して安全である。順向方式は、全てのサーバーをコンプロマイズする敵対者に対して、その敵対者があまり早くコンプロマイズしないとの条件で、安全である。この特性は、更新段階の間にサーバーが保持した秘密共有を再び無作為化することにより達成される。更新プロトコルは、秘密を、プロセス中にそれを明かさずに、再ディーリングすることに等しい。形式的には、更新は下記を達成しなければならない：

(1) 更新プロトコルに参加するサーバーの数がｋ又はそれよりも少ないグループは他のサーバーの新しい共有について知ることはできない。

(2) 前のｎ共有のうちのｋと新しいｎ共有のうちのｋとを知ることは秘密共有に関する情報を明かさない。

【００１４】［順向性対しきい値、ロバスト及び回復］敵対者は、更新段階プロトコル中は再構築段階中よりも強くてはならない。よって、ロバストで回復可能なしきい値秘密共有方式を順向化するために、更新段階は、最大ｋサーバーを作動禁止又は制御できる敵対者に対して安全でなければならない。

【００１５】秘密共有での順向性及び回復可能性はそれらが一緒に用いられるとき最も意義がある。回復機構なしに、順向秘密共有は一定のタイプの攻撃に対する保全性を高めるが、他のタイプの攻撃に対しては保全性を低くする。非順向性秘密共有方式は、記憶段階の間にサーバーを制御できる敵対者に対して安全である。なぜなら、サーバーは当該段階の間には何もしないからである。順向方式の記憶段階の間に、サーバーは周期的にそれらの共有を更新する。それゆえ、サーバーは、更新段階の間に凍結又は制御される場合、次のラウンドで適切な共有を有しないであろう。これは作動禁止する能力を有する敵対者の場合に等しい。よって、順向方式は、１つのラウンドで最大ｋサーバーを制御できる敵対者に対して、それが回復機構を有する場合にのみ、安全である。それは、１つのラウンドで最大ｋサーバーを制御又は作動禁止できる敵対者に対しても安全である。

【００１６】

【発明が解決しようとする課題】本発明の第１の目的は

順向性、ロバスト及び回復可能な分散しきい値秘密共有方式を有する方法及び装置を提供することにある。本発明の第2の目的は上記方式を用いて順向性の安全なキー証明権限を提供することにある。

【００１７】

【課題を解決するための手段】上記及びその他の目的は、サーバーのセットを用いる順向性しきい値秘密共有暗号システムを提供する方法及びシステムにより達成される。順向性秘密共有暗号システムは、ｎサーバーのうちの少なくとも(k+1) サーバーが活動状態であり且つ正直である場合にサービスが維持される点で、分散しきい値暗号システムである。秘密署名キーは、少なくとも(k+1) サーバーに敵対者が侵入する場合にのみコンプロマイズされる。それは、敵対者に侵入されたサーバーが不正をしているときでも、正直なサーバーは誤りのあるサーバーを検出し且つサービスは中断されない点でロバストである。それが回復可能であるのは、敵対者がコンプロマイズしたサーバーにある全ての局所情報を敵対者が消去する場合、サーバーは正しいプロトコルの実行に復帰すると直ちに前記情報を復元できるからである。本発明の方法及びシステムは順向性を有する。これは、敵対者が秘密を知るためには、そのアルゴリズムの同じラウンドの間に、(k+1) サーバーに侵入しなければならないことを意味する。なぜなら、秘密共有は周期的に再分散され且つ再び乱数化されるからである。２つのラウンド間の更新の間に保全必要条件を得るために、本発明は検証可能な秘密共有機構を用いる。本発明は、サーバーは同報通信媒体により通信し、それらは完全に同期され、破壊できない局所クロックを有し、そしてそれらは局所の真の乱数化のソースを有すると想定する。この方式の保全は、大きな素数順のフィールドでの対数計算の扱いにくさを想定することに依存する。それは、サーバー間の確認に用いられる”エルガマル”署名方式の保全性にも依存する。

【００１８】

【発明の実施の形態】本発明はロバスト及び回復可能性を有する順向性しきい値秘密共有方式である。この方式のモデル及び目標ならびに使用するツールが開示される。本発明を使用するキー証明権限システムも開示される。この方式の保全は、大きな素数順のフィールドでの対数計算の扱いにくさを想定することに依存する。それは、サーバー間の確認に用いられる”エルガマル”署名方式の保全性にも依存する。”エルガマル”署名方式は T. ElGamal, A Public Key Cryptosystem and a Signature scheme Based on Discrete Logarithm, IEEE Trans. on Informational Theory 31, p. 465, 1985に開示されている。

【００１９】［システムのモデル及び敵対者に関する想定］順向的に値ｘを秘密共有するｎサーバーのシステム

【数２】$A = \{P_1, P_2, \ldots, P_n\}$

【００２０】を想定する。また、このシステムは安全に且つ適切に初期化されると想定する。この方式の目標は、敵対者がｘを知ることを阻止すると同時に、サーバーＡがｘの再構築を必要とするときｘをそれら自身で再構築するのを敵対者が阻止できないようにすることである。サーバーＡ及びそれらが通信する通信ネットワークに関する仕様が与えられる。また、サーバーＡとその人的管理の間の対話機構ならびに前記管理が負う責任も指定される。

【００２１】［順向性サーバーのモデル］本発明を実施するサーバーの代表的なハードウェア構成は図１に示される。図１は、中央処理装置10、例えば通常のマイクロプロセッサ、及びシステムバス12を介して相互接続された複数の他の装置を有する、本発明によるワークステーションの典型的なハードウェア構成を示す。図１に示すワークステーションは、ランダムアクセスメモリ(RAM) 14と、読取専用メモリ(ROM) 16（図示せず）と、（ディスク装置20及びテープ装置40のような周辺装置をバス12に接続する）入出力(I/O) アダプタ18と、（キーボード24、マウス26、スピーカー28（図示せず)、マイクロホン32（図示せず）及び（又は）他のユーザインタフェース装置、例えばタッチスクリーン装置（図示せず）をバス12に接続する）ユーザインタフェースアダプタ22と、（ワークステーションを通信ネットワークに接続する）通信アダプタ34と、（バス12を表示装置38に接続する）表示アダプタ36とを含む。

【００２２】図２はサーバーＡが秘密リンクＬを介して一般的な同報通信媒体Ｃにいかに接続されるかを示す。この媒体は通信チャンネル又は通信ネットワークと呼ばれ、外部世界にも接続される。同報通信媒体は、あるサーバーにこの媒体を接続するリンクからこの媒体にメッセージが送られる毎に、この媒体に接続された全ての他のリンクにそのメッセージが瞬時に届く特性を有する。本発明は、安全で且つ同期されたクロック（図示せず）がＡのサーバーに装備され、敵対者がサーバーを制御又は作動禁止するときでも変更しないし、障害も起きないと想定する。これらのクロックは時間をラウンド及び更新段階に分割する。同期により、Ａの全てのサーバーは同時に更新段階を開始する信号を取得する。Ａの全てのサーバーは真に乱数化のソース（図示せず）を有することも想定される。もし敵対者がサーバーをコンプロマイズしても、その敵対者は当該サーバーに生成される将来の乱数を予測できないことも想定される。どのサーバーも、それらに関して以下に説明するアルゴリズム（図示せず）によるコードを含む比喩的な破壊不可能なボックスを有する。

【００２３】［敵対者］サーバーＡを攻撃する敵対者のモデルは”移動誤り（モービル・フォルト）”モデルの拡張である。このモデルは R. Ostrovsky & M. Yung, How to Withstand Mobile Virus Attacks, Proc. of th

e 10th ACM Symposium on the Principles in Distributed Computing, 1991, pp. 51-51 and R. Canetti & A. Herzberg, Crypto 94 に記述されている。ラウンドｔの間のどこかの時点で敵対者が変造するＡのサーバーの数を$k_1(t)$とし、ラウンドｔの間のどこかの時点で敵対者による攻撃が活動状態である通信チャンネルとのリンクを有する別のサーバーの数を$k_2(t)$とする。そのとき、本発明の順向性秘密共有方式は各ラウンドｔについて

【数３】 $(k_1(t)+ k_2(t)\leqq k)$

【００２４】である敵対者に対して安全である。この限界を満たすために、本発明では、Ａのサーバーの数ｎは次の式を満たす必要がある。

【数４】 $n \geqq 2k + 2$

【００２５】この方式は更に強力な敵対者に対しても安全でありうる。第１に、次の式に示すように、１少ないサーバーによるラウンドで敵対者に同じ限界ｋを得ることができる。

【数５】 $2k + 1 \leqq n$

【００２６】第２に、同期機構の変化により、リンクへの攻撃の増加がありうる。なぜなら、前記攻撃は全てシステムにより常に検出され、管理が敵対者をリンクから取り除くまでサーバーはそのプロトコルを中止しうるからである。

【００２７】サーバーの変造は下記のどれかを意味する:

・サーバーのコンプロマイズ: サーバーが記憶する全てのデータを知る。

・サーバーの制御: 指定されたプロトコルに関してサーバーに誤りを生じさせる。これは機械の凍結を含む。

・サーバーの作動禁止: そのデータを変更（又は消去）する。

【００２８】更に、敵対者は常に公開として区分されたデータを知り、そして敵対者は機械の各々が実行するアルゴリズムＭを知っている。簡略化のため、本発明は”普通”の電力中断、データ脱落及び他のハードウェア障害を敵対者の活動として処理する。よって、機械に障害がある度に、それを変造したのは敵対者であると想定する。

【００２９】２つのラウンドの間の更新段階中のどこかの時点でサーバーが変造されたとき、これらのラウンドの双方でサーバーは変造される。変造されたサーバーをカウントするこの方法の背後にある理由は、更新段階中に１つのサーバーから別のサーバーに移動する敵対者と、両者にずっと留まってだけいる敵対者とを識別することが不可能か又は非常に困難なことである。更新段階が１ラウンドの長さに較べて無視できる程に短いこの設定では、この識別はあまり重要ではない。更に、本発明は、１つのサーバーから他のサーバーに飛び移る敵対者を、それらのサーバーがどちらもそのラウンドの間ずっと変造されている場合と同じように処理する。

【００３０】Ａのサーバーに対する攻撃とは別に、敵対者はサーバー間の通信ネットワークを攻撃できる。敵対者は、そのリンクを介してサーバーＡを接続する同じ通信チャンネルＣに接続されるそれ自身のサーバーＥを有すると想定する。よって、敵対者はチャンネルＣで同報通信されるメッセージを聴取し且つメッセージをチャンネルＣに同報通信しうる。しかしながら、メッセージが通信チャンネルに届くと、敵対者は、それに接続された全てのリンクに、そのメッセージが届くのを阻止できない。敵対者は同じＡのサーバーを通信チャンネルで接続するリンクへの攻撃を行うこともできる。サーバーＰ ∈ Ａとのリンクが前記攻撃を受ける場合、敵対者は次の２つの方法で攻撃を阻止できる: １つの方法は、チャンネルＣで同報通信されるメッセージをＰが聴くのを阻止することであり、もう１つの方法は、同報通信されたＰによりチャンネルＣで送られるメッセージを停止することである。形式的には、敵対者はリンクＬ及び通信チャンネルＣの両者の全てのテープを見ることができる。あるリンクＬのアルゴリズムを敵対者のアルゴリズムと置き換えることは当該リンクへの積極的な攻撃を構成するが、チャンネルＣのアルゴリズムを変更することはこのモードでは禁止される。

【００３１】本発明は、サーバーＡ及びリンクＬを攻撃する敵対者は”除去可能”である、即ち敵対者を、それが検出されたとき、除去することができると想定する。しかしながら、サーバーＥは”除去可能”ではなく、追跡し又はチャンネルＣから遮断することはできない。

【００３２】［システムの人的管理に関する想定］Ａの全てのサーバーは、人的管理に関する情報を表示できるコンソール38を有する。マネジャは、コンソール38を介して、サーバーをリブートし、そして確認キーの更新の手順を実行する。コンソールプログラムによりデータがサーバーに入力される。管理は下記のように委ねられる: マネジャサーバーに干渉せず且つマネジャはマネジャのために指定されたプロトコルで不正をしない。換言すれば、管理はサーバーをリブートし且つ確認キーの再設定の時期及び方法に関する指示に任される。

【００３３】人的管理はリンク上の攻撃する敵対者を突き止める責任を有する。リンクが攻撃されているかどうかを検査し且つ潜在的な攻撃者を除去するように管理が指示されるときは必ず、手順は常に成功するものと仮定する。また、サーバーをリブートし、公開確認キーを再設定し、そしてリンクから敵対者を除去する全ての手順に要する時間は１ラウンドの長さよりも短いと想定する。

【００３４】［秘密共有方式］図３の流れ図は本発明の順向性検査可能秘密共有方式の良好な方法を示す。図４は図３のステップ314 の詳細な流れ図である。同様に、図５は図４のステップ410 の詳細な流れ図である。

【００３５】［初期化］ステップ310 で、サーバーは初期化される。ｐ及びｑは素数とし、ｍは２、３又は４のような小さな整数とし、次の式が成立つものとする。

【数６】$p = mq + 1$

【００３６】ｇは次数ｑのZ_pのエレメントである、即ち次の式が成立つものとする。

【数７】$g^q = 1 \pmod p$

【００３７】素数ｐは”エルガマル”暗号化及び署名方式を安全にするために選択される。秘密値ｘはZ_qに属する。限定フィールドZ_qにわたる”シャミール”の秘密共有の変形をしきい値秘密共有方式として使用する。Z_qに(k+1) 次の多項式ｆが存在し、次の式が成立つものとする。

【数８】$f(0) = x \pmod q$

【００３８】そして、全てのサーバーP_i, $i \in \{1,n\}$は次の式で示されたその秘密共有を有するものとする。

【数９】$X_i^{(1)} = f(i) \pmod q$

【００３９】指数(1) はこれらが最初のラウンドで用いた値になることを示す。

【００４０】更に、各サーバーP_iは、その秘密確認及び暗号化キー$w_i^{(1)}$乱数をZ_q内に有する。それらのキーの公開の対応部は次の式に示される。

【数１０】$\{r_i^{(1)}\}$ $i \in \{1,n\}$

【００４１】ここで、次の式は公開であり且つＡの全てのサーバーにより記憶される。

【数１１】

$$r_i^{(1)} = g^{w_i^{(1)}} \pmod p$$

【００４２】更に、あらゆるサーバーは秘密を再構築する際のロバストについて全てのｘの一方向ハッシュのセットを必要とする。本発明は一方向ハッシュとして累乗を用いるので、あらゆるサーバーは次の式で示されたセットを記憶する。

【数１２】$\{y_i^{(1)}\}$ $i \in \{1, n\}$

【００４３】ここで、次の式が成立つ。

【数１３】

$$y_i^{(1)} = g^{x_i^{(1)}} \pmod p$$

【００４４】モデルの説明で述べたように、本発明は初期化段階では（受動的にも能動的にも）敵対者が存在しないと想定するので、この初期化は公然と行いうる。

【００４５】初期化の後、Ａのサーバーの局所クロックは時を刻み始める。この時点で、複数のサーバーの動作は離散した時間間隔、即ちラウンドで同期される（ステップ312)。各ラウンドの終りで、更新段階がトリガーされる（ステップ314)。更新段階で、サーバーＡは更新プロトコルを実行する。そして管理が任意の（最大ｋまでの）キー再設置手順とそれに続く共有再構築プロトコルを実行するために予約された時間がある。更新段階はこれらの全てを達成する十分な長さを有するが、ラウンドと比較すれば短いと想定される。

【００４６】［更新プロトコル］共有を更新するために、ステップ314 及び図４に示すように、本発明は”アール・オストロフスキー”及び”エム・ユング”が提案した更新プロトコルの簡略化されたバージョンに適合する。Z_qにあるｋ次の多項式の値$f(0)$がｘに等しい場合に秘密ｘが記憶されるとき、秘密ｘはそれをｋ次の無作為の多項式$\delta(\cdot)$に加えることにより更新できる。ここで、$\delta(0)$ は０に等しいので、次の式が成立つ。

【数１４】

$$f^{(t+1)}(0) = f^{(t)}(0) + \delta(0) = x + 0 = x$$

【００４７】”オストロフスキー”及び”ユング”は、ある点で多項式を計算する演算の直線性が次の式の非常に簡単な共有更新ができることに注目した。

【数１５】$x_i = f(i)$

【００４８】即ち、次の関係式が成立つ。

【数１６】

$$f^{(t+1)} \delta(\cdot) = f^{(t)}(\cdot) + \delta(\cdot) \pmod q \quad <====>$$

$$\forall_i f^{(t+1)}(i) = f^{(t)}(i) + \delta(i) \pmod q$$

【００４９】本発明では、次の式が成立つ。

【数１７】

$$\delta(\cdot) = (f_1(\cdot) + f_2(\cdot) + \ldots + f_n(\cdot)) \pmod q$$

【００５０】次の式で示される(k+1) 次の多項式は、ｉ番目のサーバーにより独立して且つ無作為に取り上げられる。

【数１８】$f_i(\cdot)$, $f_i(0) = 0$, $i \in \{1, n\}$

【００５１】各サーバーP_iの更新プロトコル、$i \in \{1, n\}$ は下記のようになる：

(1) P_iは次の式で示される(k+1) の乱数をZ_qから取出す。

【数１９】$\{f_{ij}\}$ $j \in \{1, (k+1)\}$

【００５２】これらの数は次の式に示す多項式を定義する。

【数２０】

$$f_i(z) = f_{i1} z^1 + f_{i2} z^2 + \ldots + f_{i(k+1)} z^{(k+1)}$$

【００５３】その自由な係数は０であるので、次の式が成立つ。

【数２１】$f_i(0) = 0$

【００５４】(2) 全ての他のサーバーP_jに関して、P_iは次の式をP_jに送る。

【数２２】$f_i(j) \pmod q$

【００５５】(3) P_iは次の式で示されるその新しい共有を計算する。

【数２３】

$$x_i^{(t+1)} \leftarrow x_i^{(t)} + (f_1(i) + f_2(i) + \ldots + f_n(i)) \pmod q$$

【００５６】そしてPiは、その現在の秘密キー $x_i^{(t+1)}$ を除き、それが用いた全ての変数を消去する。

【００５７】制御能力を有する敵対者に対してこのプロトコルを安全にするために、本発明は P. Feldman, A Practical Scheme for Non-Interactive Verifiable SecretSharing, Proc. of the 28th IEEE Symposium on the Foundations of ComputerScience, pp.427-37. 1987で提案され、そして T.P. Pederson, DistributedProvers with Applications to Undeniable Signature, Eurocrypto '91, 1991で論議された一方向機能を用いて検査できる秘密共有の機構を用いる。この特定の検査可能秘密共有方式が用いられるのは、それが非対話型であり、そしてその副次効果として、それがx_iの更新と一緒に秘密の指数y_iの更新を可能にするからである。

【００５８】下記の２つの場合に、正直なサーバーは、サーバーPiにより生成された更新多項式fiを、それらが共有することを異議なく "無効" とマークすべきである。

(1) Pjが受取る共有$f_i(j)$が(k+1) 次の多項式の値ではない、又はそれらはその値であるが、この多項式fiは、$f_i(0) \neq 0$であるので、正しい更新多項式ではない。

(2) 公開値y_jの更新に用いられる共有

【数２４】

$$\{g^{f_i(j)}\}\, j \in \{1, n\}$$

【００５９】が、fiの秘密更新共有

【数２５】 $\{f_i(j)\}\, j \in \{1, n\}$

【００６０】に対応しない。

【００６１】秘密／公開キー対応部としての値w_i／r_iは、Piから来るメッセージの確認、及び "エルガマル" 暗号化によるPi宛のメッセージの暗号化に用いられる。PiがP_jについて$m \in Z_q$ を暗号化するとき、それは

【数２６】 $E_j^k[m] = (m(r_j)^k, g^k)$?

【００６２】を送る。ここで、$k \in Z_q$は乱数であり、そして指数はZ_pで計算される。受信装置は

【数２７】

$$m = m(r_j)^k * (g^k)^{(-w_i)} \pmod p$$

【００６３】を用いて解読する。

【００６４】署名動作は衝突のないハッシュ機能 $h : N \rightarrow Z_q$ を用いる。キーw_iを有するメッセージmの署名は、

【数２８】 $S_i[h(m)] = (r, s)$

【００６５】である場合に、

【数２９】

$r = g^k \pmod p$, $s = k^{(-1)}(h(m) - rw_i) \pmod q$

【００６６】である。この署名は、w_iの公開対応部

【数３０】

$$r_i = g^{w_i} \pmod p$$

【００６７】により、等式

【数３１】

$$g^{h(m)} \stackrel{?}{=} r^s\, r_i^r \pmod p$$

【００６８】を検査することにより確認できる。ハッシュ機能hは、$m' \neq m$の場合に誰もいかなる対 $m', S_i[h(m')]$ を生成できない m, $S_i[h(m)]$を知る特性を有しなければならない。

【００６９】全更新段階を通じて、暗号化及び確認は値$w_i(t), r_i(t)$ により実行される。ここでtは終了したばかりのラウンドである。このラウンド中にサーバーをコンプロマイズさせた敵対者は、このサーバーが更新プロトコルの間に署名及び解読に用いる秘密キーを知るであろう。

【００７０】[完全な更新プロトコル] 下記のステップはステップ314 でサーバーが用いた更新プロトコルの詳細である。

【００７１】ステップ410 で、各サーバーPiは、Z_qで更新多項式

【数３２】 $f_i(z) = f_{i1}z^1 + f_{i2}z^2 + \ldots + f_{i(k+1)}z^{k+1}$

【００７２】を定義するZ_qにある(k+1) の乱数$f_{i1}, f_{i2}, \ldots, f_{i(k+1)}$ を取出す（ステップ510)。それは、この多項式の(n-1) の共有

【数３３】 $u_{ij} = f_i(j) \pmod q$, $j \neq i$

【００７３】を形成し（ステップ512)、そしてそのメッセージmsg:

【数３４】

$$(g^{f_{i1}}, g^{f_{i2}}, \ldots, g^{f_{i(k+1)}}, E_1^{k_{i1}}[U_{i1}]\, E_2^{k_{i2}}[U_{i2}], \ldots, E_{(i-1)}^{k_{i(i-1)}}[U_{1(i-1)}]\, E_{(i-1)}^{k_{i(i+1)}}[U_{j(i+1)}], \ldots, E_n^{k_{in}}[U_{in}]) \quad (1)$$

【００７４】を生成する。ここで、全ての指数がZ_pで計算される（ステップ514)。更に、各サーバーPiは、Z_q内でその新しい秘密キー$w_i^{(t+1)}$ を乱数として取出し（ステップ516)、そしてその対応する新しい公開キー:

【数３５】

$$r_i^{(t+1)} = g^{w_i^{(t+1)}} \pmod p$$

【００７５】を計算する（ステップ518）。そしてPiは、その古いキーwi(t) により署名された対:

【数３６】 (msgi, ri(t+1))

【００７６】を同報通信する（ステップ412 及び520）。

【００７７】ステップ414 で、各サーバーPiは、それが前のステップで受信したメッセージを考慮する。あるjで、それが形式

【数３７】 (msgj, rj(t+1))

$$g^{u_{ji}} \stackrel{?}{=} (g^{f_{j1}})^{i} (g^{f_{j2}})^{i^2} \cdots (g^{f_{j(k+1)}})^{i^{(k+1)}} \pmod p \quad (2)$$

【００７８】の確認されたメッセージを受信しないか又は２つ以上受信した場合、それはPjを ”無効” とマークする。Biは ”無効” とマークされなかったサーバーの索引のセットとする。そしてPiはメッセージmsgj, j ∈ Biの

【数３８】

$$E_i^{k_{ji}}[u_{ji}]$$

【００７９】部分を解読する。そして、あらゆるj ≠ iについて、それはメッセージmsg:

【数３９】

はリンク攻撃が活動状態である場合に管理に追跡させる。しかしながら、新しい公開キーの肯定応答の欠如はBiに影響しない。あるサーバーPj, j∈Biによる告発 accjiの各々について、サーバーPiは、Pjとの通信で用いられた共有及び乱数ベクトルを含む署名された応答:

【数４２】 respij = (i, j, kij, uij)

【００８５】を同報通信するので、誰が不正をしていたかを決定するための公開審査を可能にする。これらの応答は１つの署名と連結され、同報通信されるべきである。

【００８６】ステップ420 で、同じサーバーの署名を有する２つ以上の応答が同報通信された場合、全てのサーバーPはそれを ”無効” とマークし、そのセットBiを変え、そして管理のメッセージを表示する。各サーバーPは、ステップ(1) でPiにより適切に同報通信された（accji, respij）の全ての対についてそれ自身の決定を行う。respijで送られた値は疑わしいとみなされるので、それらは次の式に示す ”素数” により表示される:

【数４３】

$$k'_{ij}, u'_{ij}$$

【００８７】あらゆるサーバーPは下記のアルゴリズムにより決定する:

(a) もし

【数４４】

$$E_j^{k_{ij}}[u_{ij}] \neq (u'_{ij}(r_j^{(t)})^{k_{ij}'} \pmod p, g^{k_{ij}'} \pmod p)$$

【数４５】

$$u'_{ij} = u_{ij}$$

【００８９】は多項式fiの正しい共有であるかどうかを検査する。この等式が正しくない場合、それはPiが無効な共有をPjに送ったことの証明であるので、それは ”無

【００８８】であれば、即ち PiがPjの告発に全く応答しなかったとき、Piは不正をしているので、それは ”無効” とマークされなければならない。さもなければ、

(b) もし (uij,kij)が実際に msgi で用いられたならば、ステップ(2) でPjが用いるべき同じ等式(2) を評価することにより、

効"とマークされなければならない。等式が正しい場合、PはPjを"無効"とマークする。

【００９０】"無効"とマークされた各サーバーについて、セットBiは変えられ、そして対応するメッセージが

$$x_i^{(t+1)} \leftarrow x_i^{(t)} + \sum_{k \in B_i} \left[f_k(i) \right] \pmod{q}$$

$$\forall_{j \neq i} \left[y_j^{(t+1)} \leftarrow y_j^{(t)} * \prod_{k \in B_i} g^{f_k(j)} \pmod{p} \right]$$

【００９２】各サーバーは新しい共有$x_i^{(t+1)}$、新しいキー$w_i^{(t+1)}$、そして新しい公開キーのセット

【数４７】 $\{y_j^{(t+1)}\}$ $j \in \{1,n\}$

【００９３】及びセット

【数４８】 $\{r_j^{(t+1)}\}$ $j \in \{1,n\}$

【００９４】を除いて、このプロトコルで用いた全ての変数を除去する。

【００９５】正しい更新多項式$\delta(\cdot)$は合計

【数４９】

$$\sum_{k \in B} \left[f_k(\cdot) \right] \pmod{q}$$

【００９６】に等しい。ここで、少なくとも（k+2）のサーバーPi, i ∈ {1,n}についてB =Biである（ステップ420)。

【００９７】サーバーPが既知の形式のメッセージの署名された同報通信の受信を予期しており、そしてそれが２つの前記メッセージを受信し、どちらも同じサーバーPiから来るらしいときは必ず、PはPiを無効とマークする。もちろん、攻撃者はPiが行ったものと同じメッセージを常に送信できる。このプロトコルでは、全てのサーバーは最大３つのメッセージまで送信する。それゆえ、Pは、Piが署名したように見えるメッセージのセットCiを受取るときは必ず、該プロトコルのどれか前のステップでPiにより同報通信されたCiのエレメントを捨てるべきであり、そしてCiの全ての他のエレメントが同じである場合、このエレメントをPiのメッセージとみなす。さもなければ、Piを"無効"とマークすべきである。なぜなら、それは不正をしているか、又はその確認キーをコンプロマイズした敵対者により複製されているからである。別の良好な点は、Piがそれ自身が署名したように見えるメッセージを調べる場合、それはそれ自身も"無効"とマークし、従ってBjを計算することである。このように、たとえそれが複製されていても、それはその新しい共有$x_i^{(t+1)}$ 及び全ての

【数５０】 $\{y_j^{(t+1)}\}$ $j \in \{1,n\}$

【００９８】を正しく計算する。その新しい公開確認及び暗号化キー$r_i^{(t+1)}$ は肯定応答されないので、気を付けなければならない。

【００９９】［更新プロトコルの分析］敵対者は次の３つの方法で上記プロトコルに干渉することができる：

Piのコンソールに表示される。

【００９１】ステップ422 で、各サーバーPiは次の式を計算する：

【数４６】

・ 敵対者はこのプロトコルでその敵対者が管理するサーバーに不正をすることができる。使用禁止されたサーバーは（それが間違ったキーでメッセージを暗号化し且つ署名する場合）それが不正をしているかのように見えるか又はそれが乱数メッセージを送っているかのように見える。

・ 敵対者はその敵対者がこの更新段階前のラウンド中にコンプロマイズしたサーバーのキーにより署名されたメッセージを送信できる。他のサーバーについては、このケースはサーバーがなお敵対者により制御されるときのケースと区別できない。

・ 敵対者はサーバー及び通信チャンネルの間のリンクを攻撃できる。

【０１００】全ての更新ラウンドで、正直であり、前のラウンドでコンプロマイズされておらず、通信チャンネルとのリンクに対する攻撃は活動状態ではない、(K+2)のサーバーがあるから、これらのサーバーのセットBiは全て同じであり、そして少なくとも(k+2) のエレメントを有する。それゆえ、秘密共有多項式ｆは少なくとも(k+2) の多項式fiにより更新され、敵対者はそれらをどれも再構築できない。最悪の場合、敵対者はこれらの多項式の各々からｋの共有を学習できるが、それらは(k+1)次の全てであり且つそれらの各々について $f_i(0) = 0$ であるから、敵対者はそれらを再構築するためには(k+1) の共有を必要とする。

【０１０１】［更新後の回復］サーバーのコンソールに表示されたメッセージから、管理は、敵対者によりどの機械が制御され、コンプロマイズされ又は使用禁止され、そしてどのリンクに対する攻撃が活動状態であったかを決定できる。特に、コンソールは下記の２つのセットの（しばしば交差する）サーバーに情報を表示する：

・第１のセットはそれらの新しい公開キーの幾つかの肯定応答を受信しなかったサーバーである。

・第２のセットはそれらの新しい公開キーが少なくとも(k+2) のサーバーにより肯定応答されず、それらの秘密共有$x_i^{(t+1)}$ が間違って計算され、そして新しい多項式 $f^{(t+2)}$ と一致せず、更に、次の式に示すそれらの現在のセットのビューが正しくないサーバーである。

【数５１】 $\{y_j^{(t+1)}\}$ $j \in \{1,n\}$

【０１０２】第１のセットは活動状態のリンク攻撃に関

する管理情報を与える。そしてマネジャは導き出されたリンクから迅速に敵対者を取り除く。第２のセットに関して、マネジャは最初にキー再設定手順（ステップ316）を実行し、そして自動共有回復プロトコル（ステップ318)を開始させる。公開キー{r_j}が良好に分散され且つ他のサーバーにより肯定応答されているＡのサーバーのセットをＣとする。そして次の式が成立つものとする。

【数５２】$|C| \geqq k+2$

【０１０３】P_iのキーの再設定（ステップ316)は次にように行われる：P_iのコンソールにより、マネジャは{r_j}$_{j\in C}$を設定し、そしてP_iにその新しい乱数の秘密キー $w_i \in Z_q$を取出すように求める。P_iはこのような数を取出し、そして次の式に示す対応する公開キーを表示する。

【数５３】

$$r_i = g^{|w_i|} \pmod p$$

【０１０４】そしてマネジャは全ての他のサーバーにこの数を設定する。セットＣはP_iにより増加される。この再設定が終了すると、マネジャはサーバーＣに関する回復プロトコルを開始させる。あるいは、サーバー自身は、ある公開キーがそれらに再設定されるときは必ず、回復プロトコルを自動的にトリガーするためにそれらが同意プロトコルを開始するようにプログラムされる。そのとき、Ｃにある全てのサーバーは使用可能である。

【０１０５】［共有回復プロトコル］P_uは再設定を必要とする、次の式の共有を有するサーバーを示す。

【数５４】$x_u = f(u)$

【０１０６】最初に、ステップ610 で、サーバーは(k+2）のサーバーの、次の式の初期セットを選択する。

【数５５】$B \subset C \setminus \{P_u\}$

【０１０７】それらはP_uに対するｑ自身の確認又はその逆の確認をしなければならないのでＣから取出される。これは容易である。なぜなら、サーバーは索引により順序付けられ、そしてそれらはＣの現在の状態、即ち動作中の確認及び暗号化キーを有するサーバーを知っているからである。該プロトコルは敵対者が存在するとき下記の特性を有しなければならない：

・ P_u は x_u のみを学習する、即ち、$i \in B$ である場合、他の共有 x_i を学習できない。

・ $i \in A \setminus F$である場合、(k-1）のサーバーのグループ

【数５６】$F \subset A \setminus \{P_u\}$

【０１０８】はxuを又はいかなる x_iも学習できない。

・ サーバーP_uは適切なセット {y_i}$_{i\in\{1,n\}}$を学習する。

【０１０９】ｆの有効な共有を有する(k+2）のサーバーの基本セットは

【数５７】$x = f(0) \pmod q$

である場合だけでなく、ｆの任意の他の値、特にP_uが必要とする共有

【数５８】$x_u = f(uv) \pmod q$

【０１１０】も回復できる。これはラグランジェ補間法の公式で行うことができる。協同することになっている(k+2）のサーバーのセットＢが設定された後、各P_j，$i \in B$はP_uに

【数５９】

$a_i = x_i \Pi_{j\in B, j\neq i}[(u-j)/(i-j)] \pmod q$

【０１１１】を送る。そしてP_uはそれらを加え合わせて

【数６０】

$$x_u = f(u) = \Sigma_{j\in B} a_i$$

【０１１２】を取得する。しかしながら、本発明がそれのみを行った場合、P_uは全ての x_iを a_i からも学習するであろう。しかし、本発明は a_i を x_v の些細な秘密共有として処理できる。それゆえ、本発明はＢのサーバーがこれらの共有を再乱数化した後にそれらを P_u に送ることを必要とする。Ｂの各サーバーP_iはZ_qにある(k+2）の乱数 {C_{ij}}$_{j\in B}$ を取出す。そして、それらはこれらの値を対で交換する:P_iはc_{ij} をP_jに与え、そしてc_{ij} を取得する。次に、それぞれが

【数６１】

$$a_i' = a_i + \Sigma_{j\in B} c_{ji} - \Sigma_{j\in B} c_{ji} \pmod q$$

【０１１３】をP_uに送る。

【数６２】

$$\Sigma_{i\in B} a_i' = x_u \pmod q$$

【０１１４】であるから、これはx_uのもう１つの秘密共有である。ここで基本方式は更新段階プロトコルで用いた同じツールを用いて検査できなければならない。

【０１１５】この方式を検査可能にする完全なプロトコルを以下に記述する。全ての同報通信は、現時点でセットＢにないものを含むＣのサーバー及びサーバーP_uを必要とする。明確化を図るために、下記は、全ての確認された同報通信について、受信者はメッセージの署名を検査し、そして（送信者が不正をしているか又はそのキーがコンプロマイズされ複製されているために）複製の攻撃があるとき、それらはこのサーバーを”無効”とマークし、それらの管理のコンソールに適切なメッセージを表示することに言及しない。これは更新プロトコルと同じ手順である。この方式を検査可能にするプロトコルは:Ｂの全てのP_iがZ_qにある(k+2）の乱数値 {c_{ij}}$_{j\in B}$ を計算し、そして:

【数６３】

$$(\{g^{c_{ij}} \pmod p\}_{j\in B}, \{E_j^{k_{ij}}[C_{ij}]\}_{j\in B}) \qquad (3)$$

【０１１６】及びこのメッセージの署名を同報通信する（ステップ612)。

【０１１７】ステップ614 で、Bの全てのP_iは上記同報通信から値 $\{c_{ji}\}_{j\in B}$ を解読し、そしてそれらがP_jにより同じメッセージで同報通信された指数

【数６４】

$$g^{c_{ji}}$$

【０１１８】を取ることにより、全てのサーバーP_jから正しい共有を与えられたかどうかを検査する。この値が一致しない場合、P_iはP_jの告発

【数６５】$acc_{ij} = (i, j, S_i[i, j])$

【０１１９】を全てのサーバーに同報通信する（ステップ616)。それはP_jも"無効"サーバーとマークする。

【０１２０】ステップ618 で、更新プロトコルと全く同じように、あるサーバーP_jによる告発acc_{ji}毎に、サーバーP_iは署名 $S_j[j,i]$ を検査し、そして告発の原告がxを知っていることが証明された場合、P_iは、P_jとの通信で用いられたc_{ij}及び乱数ベクトル:

【数６６】$resp_{ij} = (i, j, k_{ij}, c_{ij})$

【０１２１】に応答し、誰が不正をしていたかを決定する公開審理を可能にする（ステップ620)。

【０１２２】ステップ622 で、(Bのサーバーのみならず)全てのサーバーPは、２つの前のステップで同報通信された全ての対 $acc_{ji}, resp_{ij}$ について、これらが P_i及びP_jの間の通信で用いられた真の値であるかどうかを検査し、そしてそれらが真である場合、送られた値の指数

【数６７】

$$g^{c_{ji}} \pmod p$$

【０１２３】がそのセットにあるP_iにより同報通信された指数

【数６８】

$$\{g^{c_{ij}} \pmod p\}_{j\in B}$$

【０１２４】と一致するかどうかを検査する。

【０１２５】このステップは更新プロトコルのステップ420 での検査に等しい。P_i又はP_jが不正をしていると分かり、そしてPにより"無効"とマークされる。

【０１２６】ステップ624 で、あらゆるサーバーは、それが前のステップで"無効"とマークしたその局所セットF_iのサーバーを考慮する。

【数６９】$|F_i| = m \neq 0$

【０１２７】である場合、それらは新しいB:

【数７０】

$$B \leftarrow (B \setminus F) \cup \{\max(B)+1, \ldots, \max(B)+m\}$$

【０１２８】を計算する。同報通信により、正直で活動状態の全てのサーバーは同じビューのFを有する、従って同じビューの新しいBを有する。それらはBを再計算したのちステップ610 でプロトコルを再開始する。

【０１２９】ステップ626 で、前のステップで誰も不正をしなかった場合、全てのサーバーP_iはその下位共有:

【数７１】

$$a_i' = a_i + \sum_{j\in B} c_{ij} - \sum_{j\in B} c_{ji} \pmod q \qquad (4)$$

【０１３０】を計算する。ステップ628 で、Bの各サーバーP_iは全てのサーバーに

【数７２】

$$E_{r_u}^{k_l}[a_i']$$

【０１３１】を同報通信する。ここで、k_iはZ_qにある乱数である。

【０１３２】ステップ630 で、P_uはこれらの同報通信から全ての

【数７３】

$$\{a_l'\}_{l\in B}$$

【０１３３】を解読し、そして

【数７４】

$$x_u = \Sigma_{l\in B} a_l' \pmod q$$

【０１３４】を計算する。そして、それは、この値の指数

【数７５】

$$g^{x_u} \pmod p$$

【０１３５】が、最初にP_uで設定された公開キー y_uと同じであるかどうかを検査する。同じである場合、再構築はステップ632 で終了する。さもなければ、Bからのサーバーのなかには不正をしていたものがあるに違いない。Pは全ての $i \in B$について:

【数７６】

$$g^{a_i} \leftarrow (y_i)\Pi_{j\in B, j\neq i}[(v-j)/(i-j)] \pmod p$$

$$g^{a_i} \stackrel{?}{=} g^{a_i} * \prod_{j\in B} g^{c_{ij}} * \left(\prod_{j\in B} g^{c_{ij}}\right)^{(-1)} \pmod p$$

【０１３６】を評価することにより不正をした者を見つける。ここで、値

【数７７】

$$\{g^{c_{ij}}\}_{i,j\in B}$$

【０１３７】はステップ612 で同報通信から取出される。２番目の式が適用できない場合、それはP_iが正しい a_i' を P_uに送らなかったことを意味する。それを他に対して証明するために、P_uはその告発

【数７８】$acc_i = (i, S_{temp}[i])$

【０１３８】を全ての他のサーバーに同報通信する（ス

テップ634)。また、それは不正をしたものをそれ自身で全て"無効"セットFとマークする。

【0139】ステップ636で、このプロトコルのステップ618におけるように、全ての告発されたサーバーP$_i$は、告発acc$_i$の署名を検査し、もしそれが有効であれば、次の式に示す応答を同報通信する。

【数79】

$$resp_i = (i,\ k_i,\ a_i')$$

【0140】ステップ638で、全てのサーバーPは全てのこれらの告発を検査するに当り、最初にこれらが通信で用いた値であることを検査し、次に上記の実行されたP$_u$と同じ検査式を検査する。更に、それらは全てそれらが無効であると検出した同じサーバーのセットFで到来する。不正をしているのはP$_u$であることが判明した場合、それらは全て適切なメッセージをシステムマネジャに送り、そして回復手順は停止される(ステップ640)。この事象はP$_u$が敵対者により再び制御されることを意味する。これは他の敵対者検出と全く同様である。システムマネジャはサーバーが表示した警告を読取り、そして全回復プロセスを再開始する。しかしながら、P$_u$の告発が正しい場合、通常のように、サーバーはシステム管理のための適切な警告を表示し、そしてセットB:

【数80】

$$B \leftarrow (B \setminus F) \cup \{\max(B) + 1, \ldots, \max(B) + m\}$$

【0141】を再計算することにより回復を続ける。ここで、

【数81】$|F| = m$

であり、そしてステップ610から全プロトコルを再開始する。

【0142】[制御されたサーバーが存在するときの更新プロトコルの保全]不正をするkのサーバーが存在するときの上記プロトコルの保全及びリンクに対する攻撃がないことは、検査可能な秘密共有に変えられる。即ち、解決は、どれか特定のサーバーP$_i$がそのプロトコルで不正をしていたかどうかを正直なサーバーP$_j$が検出できる機構である。このケースでは、それらは異議なくその更新多項式f$_i$の共有を"無効"とマークする。上記プロトコルでP$_i$が不正を行うのは、それがステップ612で他のサーバーに送る値がZ_qで、

【数82】$f(0) = 0 \pmod q$

が成立つ、任意のk次の多項式fの正しい値ではない場合、そしてその場合のみである。これは、検査された値の共有と呼ばれる、検査可能な秘密共有の幾分か簡略化されたケースである。なぜなら、全てのサーバーPは(n-1)の他のサーバーの間に既知の値0を分散する(それはそれ自身にも1つの共有を与える)。検査可能な秘密共有と全く同じように、正直なサーバーは、"秘密の"値0がPにより正しく共有されたかどうかに異議なく同意するであろう。また、検査可能な秘密共有と全く同じように、それは、各サーバーがそれ自身の秘密共有のみを学習するように行われるべきである。

【0143】変更された更新プロトコルのステップ618～620は上記から:

【数83】

$$x_i^{(t+1)} \leftarrow x_i^{(t)} + \Sigma_{k \in B_i} \lceil f_k(i) \rceil \pmod q$$

【0144】である。ここで、B_iは、更新多項式がP$_i$により"無効"とマークされなかった全てのサーバーの索引のセットである。検査可能な秘密共有機構により、更新段階の間にP$_i$及びP$_j$がどちらも正直である場合、

【数84】$B_i = B_j$

【0145】である。即ち、他のサーバーの正直さについてそれらは同じ判断を有する。

【0146】既存の検査可能な秘密共有のプロトコルは全て同報通信チャンネルを必要とする。よって、この点から、nの秘密共有サーバーの各々はそれを普通の同報通信チャンネル、エサーネットに接続するリンクを有すると想定する。この同報通信チャンネルに乗る全てのメッセージは、このチャンネルをサーバーに接続する全てのリンクに到着する。

【0147】更新の間に最大kの不正をしている、私用禁止、即ち凍結されたサーバーが存在するときの検査可能な秘密共有が達成される場合、全ての正直なサーバーがその共有を更新する、少なくとも(n-k)の更新多項式f$_i$がある。

【0148】共有の再乱数化の次数は順向方式の第2の特性、即ち更新前及び更新後にkの共有を知ることは敵対者に秘密を再構築させないことを保証するのに十分である。以下はkサーバーが更新中に制御されるときの微妙な点である。敵対者はそのとき各更新多項式f$_i$のkの共有を知り、そしてこれは敵対者がそれらの全てを再構築することを許すのは、それらの全ての自由な係数が0であることが知られているからである。よって、敵対者は全てのサーバーP$_i$の全更新値

【数85】

$$\delta(i) = (f_1(i) + f_2(i) + \ldots + f_n(i)) \pmod q$$

【0149】を計算できる。しかしながら、更新段階の間に敵対者がkサーバーのセットFをコンプロマイズする場合、敵対者はこの更新の前又は後のラウンドで他のサーバーをどれもコンプロマイズすることができない。よって、敵対者は全ての

【数86】$\{x_i^{(t)}\}_{i \in F}$

【0150】及び

【数87】$\{\delta(i)\}_{i \in \{1,n\}}$

【0151】を知る。後者は、特に敵対者が、どのみち知る

【数88】$\{x_i^{(t+1)}\}_{i \in F}$

【０１５２】を計算することを許す。しかし、秘密の再構築を敵対者に許す追加の共有

【数８９】

$$x_i^{(t)},\ j \notin F$$

【０１５３】又は

【数９０】

$$x_i^{(t+1)},\ j \notin F$$

【０１５４】を敵対者は依然として学習できない。

【０１５５】［"重複"攻撃及びリンク攻撃が存在するときの更新プロトコルの保全］敵対者がラウンドｔでサーバーＡをコンプロマイズし、そして同じラウンドでサーバーＡから去る場合、ラウンドｔ及びラウンド(t+1) の間の更新段階中に、このサーバーは無効ではないが、敵対者は全てのその秘密を知る。（活動状態のプロトコルの混乱がない）コンプロマイズされたサーバーのみを有する敵対者を想定する。ラウンドｔの間及び更新 t/(t+1)の間に敵対者によりコンプロマイズされたサーバーのセットをＦとする。最悪のケースでは、

【数９１】$|F| = (k-1)$

である。そのラウンドの間及び更新の間に敵対者に対して絶対に安全である "有効な" サーバーのセットをＧとする。明らかに、それらがサーバーＦに送る各f_iは、 $i \in G$である。さて、プロトコルにより、それは、Ａが取得するf_i, $i \in G$ の共有を学習することもあり、しないこともある。サーバーが既知のキーで暗号化されたばかりのそれらの更新多項式の共有をＡに送る場合、敵対者はそれらを学習できる。

【０１５６】［順向性安全公開キー証明権限］順向性安全キー証明権限センターを実現するために順向性秘密共有を用いる方法を示す完全な解決が提案される。メッセージの "エルガマル" 署名動作を実行し且つ署名キーを順向的に維持するシステムが記述される。このようなシステムはキー証明権限として有効に用いうる。なぜなら、それが署名するメッセージは公開キー及び他のユーザの識別タグにありうるからである。

【０１５７】［エルガマル署名の分散バージョン］最初に、互いに署名センターを形成する参加サーバーのグループにより署名動作が実行される、"エルガマル"署名アルゴリズムの分散バージョンを想定する。署名はこれらのサーバーの部分的な署名から成る。それは１つの公開検査キーにより検査できる。これは、概念的に、単一の署名キーもあることを意味し、それは署名センターを形成するサーバー間で共有されるだけである。このアルゴリズムは、それを共有するサーバー間のこの秘密キーの分散の変化を許す、即ち、それは共有の順向性更新を許す。

【０１５８】［初期化］大きな素数をｐとし、そしてｐよりも小さい乱数をｇとする。ｐ及びｇはどちらもネットワーク内の全ての当事者が知っている。各サーバーP_iはその秘密キーを乱数 x_iとみなす。ここで、$i \in \{0, n\}$ 且つ $x_i < p-1$である。全てのサーバーの秘密キーが初期化された後、その結果生ずる全てのラウンドで、次の式の合計が得られる。

【数９２】$x_1^{(t)} + x_2^{(t)} + \ldots + x_n^{(t)} = x \pmod q$

【０１５９】ここで、$x_i^{(t)}$はラウンドｔでのサーバーP_iの秘密キーを表わし、そしてｘは署名センターの概念的な一定の秘密キーを表わす。最初に、各サーバーはその秘密キーの公開対応部:

【数９３】

$$y_i = g^{x_i} \pmod p$$

【０１６０】を計算する。そして、それらはそれらの公開キーの部分を互いに送出するので、ｘの公開検査対応部:

【数９４】

$$y \leftarrow y_1 y_2 \cdots y_n = g^{x_1} g^{x_2} \cdots g^{x_n} = g^{x_1 + x_2 + \ldots + x_n} \pmod p \quad (5)$$

【０１６１】を計算できる。シーケンス(p, g, y, n)はこの署名センターで生成された署名の検査に用いる公開キーである。

【０１６２】［署名の発行］メッセージｍに署名するために、各サーバーはそれ自身の秘密乱数 k_i、(p-1)に関連する素数を取出す。そして、各サーバー P_i はその署名の第１の部分:

【数９５】

$$r_i = g^{k_i} \pmod p$$

【０１６３】を計算する。各サーバーはそのr_iを他のサーバーに同報通信するので、それらの各々は、次の式に示すように、その署名の第２の部分を計算する:

【数９６】

$$s_i = k_i^{-1}(m - x_i r_1 r_2 \cdots r_n) \pmod{p-1} \quad (6)$$

【０１６４】最初の "エルガマル" 署名方式と全く同じように、

【数９７】

$$k_i^{-1} \pmod{p-1}$$

【０１６５】の値は、ユークリッドのアルゴリズムにより見出だすことができる。署名されたメッセージは下記のシーケンスを有する:

【数９８】

$$S(m) = (m, s_i, s_2, \ldots, s_n, r_1, r_2, \ldots, r_n) \quad (7)$$

【０１６６】［署名の検査］署名の検査を必要とする当事者は下記の式が真であるかどうかを検査する:

【数９９】

$$g^{rm} = y^{r_1 r_2 \cdots r_n} r_1^{s_1} r_2^{s_2} \cdots r_n^{s_n} \pmod p \quad (8)$$

【０１６７】これは下記の変換により真であると示すことができる:

【数１００】

$$y^{r_1 \cdots r_n} r_1^{s_1} \cdots r_n^{s_n} = (g^{x_1+\cdots+x_n})^{r_1 \cdots r_n} (g^{k_1})^{k_1^{-1}(m-x_1 r_1 \cdots r_n)} \cdots$$

$$(g^{k_n})^{k_n^{-1}(m-x_n r_1 \cdots r_n)} = g^{x_1 r_1 \cdots r_n + \cdots + x_n r_1 \cdots r_n} g^{m-x_1 r_1 r_2 \cdots r_n}$$

$$* g^{m-x_n r_1 \cdots r_n} = g^{n \neq m} \pmod p$$

【０１６８】［保全分析］上記の基本的な分散"エルガマル"アルゴリズムの保全は、本発明のプロトコルの残りの部分と同様に、大きな素数順の最後のフィールドにおける対数の計算は見込まれた多項式の計算時間から実行不可能であるとの想定による。これは"エルガマル"及びＤＳＡ暗号及び署名方式の基本的な保全を想定する。上記の"エルガマル"の公開キーアルゴリズムの分散バージョンの保全は最初のバージョンと全く同じである。攻撃者が本発明のアルゴリズムを破壊できる場合、攻撃者は通常の"エルガマル"アルゴリズムも破壊できるであろう。攻撃者はm、S(m)及び$x_1, x_2, \ldots, x_{n-1}$を知っていると想定する。攻撃者が更に必要とするものはx_nである。攻撃者は

【数１０１】

$$g^{x_n} = y * g^{-(x_1+x_2+\cdots+x_{n-1})}$$

【０１６９】を計算できる。ここで、負の指数はモジューロ(p-1)が計算される。

【数１０２】$x = x_n s_n$, $k = k_n r_1 r_2 \ldots r_n$

【０１７０】が代入される。そして攻撃者は

【数１０３】

$$g^x = (g^{x_n})^{s_n} \pmod p, \quad g^k = (g^{k_n})^{r_1 r_2 \cdots r_n} \pmod p$$

【０１７１】及び

【数１０４】

$x + 1 = s_n k_n + x_n r_1 r_2 \ldots r_n = m \pmod{p-1}$

【０１７２】を知る。この３つの式のセットで、最初の２つのうちの１つは残りの部分から取出すことができる。例えば、

【数１０５】$g^k = g^{x+k} (g^x)^{(-1)} \pmod{p-1}$

【０１７３】である。よって、残るのは下記の２つの式である。

【数１０６】

$g^x = a \pmod p$, $x + k = b \pmod{p-1}$

【０１７４】最初の"エルガマル"アルゴリズムの保全は、kが未知である場合、第２の式はxに関する情報を出さないことに基づくので、"エルガマル"アルゴリズムは見込まれる多項式の計算時間で対数

【数１０７】$x = \log_g a \pmod p$

【０１７５】を計算することは実行不可能であるのと同程度に安全である。

【０１７６】［順向性しきい値秘密共有方式への分散エルガマルの適応］本発明は"シャミール"の多項式秘密共有を上記分散"エルガマル"アルゴリズムに組込む。この組込みはしきい値順向方式を生成するが、このしきい値はラウンド中にのみ存在する。前述のように、更新段階中のしきい値は回復機構を必要とする。この方式はロバストを達成しない。よって、この予備的な方式は:

・各ラウンドで最大ｋサーバーをコンプロマイズでき、

・各ラウンドで最大ｋサーバーを凍結するが、更新段階では凍結できない敵対者に対して安全である。

【０１７７】本発明は、Threshold Cryptosystems Crypto 89, pp. 307-15 に記述された、指数のしきい値秘密共有に関する"デスメド"及び"フランケル"の解決に関連する。彼らの着想は、ｋ次の多項式による"シャミール"のしきい値秘密共有を用い、そして各サーバーで局所的にラグランジェの補間式の成分を計算することにより(k+2)の正直なサーバーのグループによる秘密の指数を計算することであった。それらの方式及び本発明の間の類似性は、多項式により共有された秘密のラグランジェ再構築を両者が用いることであるが、プロセス中に再構築の秘密を誰も学習できない。秘密は("デスメド"及び"フランケル"のケースでは)メッセージを指数化するか又は(本発明では)"エルガマル"により取出された署名を発行するためにのみ用いられる。本発明はｎサーバー間で $f(0) = x$ を秘密共有するためにｋ次の多項式ｆを用いる。それはｐ及びエレメント $g \in Z_p$ を取出し、ｇが素数順序ｑになるようにする、即ち

【数１０８】$g^q = 1 \pmod p$

【０１７８】である。Ｑはできるだけ大きくすべきであり、

【数１０９】$p = mq + 1,\ m \in \{2,3,4\}$
【０１７９】が成立つようにする。ここで、公開キーはシーケンス (p,q,g,y,k)になる。
【０１８０】本発明は、初期化段階の間に、各サーバーがそのｘの秘密共有
【数１１０】$x_i = f(i)$

$$a_i = x_i * \prod_{j \in B, j \neq i} [(0-j)/(i-j)] \pmod q \qquad (9)$$

【０１８２】を計算する。ラグランジェの補間公式から、これは
【数１１２】

$$\sum_{i \in B} a_i = x \pmod q \qquad (10)$$

【０１８３】であることを保証する。なぜなら、
【数１１３】$\forall_i,\ x_i = f(i),\ x = f(0)$
【０１８４】であり且つｆはZ_qでｋ次の多項式であるからである。
【数１１４】$(i - j),\ i \neq j$
の逆数の存在は、ｑが素数であることにより保証される。それらの部分的な秘密としてa_iを用いることにより、Ｂのサーバーは、上記:
【数１１５】

$$g^{(k+2)*m} = y^{\prod_{(r_l, s_l) \in S(m)} r_l} \prod_{(r_l, s_l) \in S(m)} r_l^{s_i} \pmod p \quad (11)$$

【０１８７】［署名プロトコルにおける検査及びロバスト］上記予備的なしきい値方式に付加しなければならない第１の特性はラウンドの間のロバストである。これは、最大ｋサーバーまで制御できる敵対者に対してそれを安全にするが、それはそのラウンドの間のみであって、更新段階の間ではない。上記しきい値方式は署名者従属である。なぜなら、署名を発行する前に、正直なサーバーは、そのメッセージに署名するために協同する、(k+2) のサーバーのセットＢに同意する必要があるからである。幾つかの現に活動状態である、正直なサーバーのセットＢを取出しうるためには、正直なサーバーが不正をする者を取り除きうる機構が必要である。最後に、B, m及び S(m) を知る各サーバーは、全てのサーバー$P_i,\ i \in B$ について、同じ m, g, p, q 及び式(5) により計算された正しい秘密キーa_iを有するプロトコルにより計算されるという意味で、その部分的な署名 $(r_i, s_i) \in S(m)$ が正しいかどうかを検査しうるべきである。
【０１８８】サーバー間のこの互いに部分的な検査を可能にするために、本発明は各サーバーの秘密共有$x_i(t)$をそれらの秘密署名キーとして処理し、そして全ての他のサーバーにそれらのキーの公開検査対応部:
【数１１７】

$$y_i(t) = g^{x_i(t)} \pmod p$$

【０１８９】を与える。全ての正直なプロセッサＰ（現にＢにあるものだけではない）はＢの全てのサーバーP_iの部分的な署名:
【数１１８】$(r_i, s_i) \in S(m)$
【０１９０】を次の２つのステップで検査する。
(1) 現在のＢ及び
【数１１９】

$$y_l = g^{x_l} \pmod p$$

【０１９１】を知り、Ｐは
【数１２０】

$$g^{a_i} = (y_l)^{\prod_{j \in B\ j \neq i} [(0-j)/(i-j)] \pmod q} \pmod p$$

【０１９２】を計算する。
(2) 上記のように
【数１２１】

$$g^{a_i}$$

【０１９３】を計算したのち、Ｐは S(m) を取出し、そ

して式:

【数１２２】

$$g^{m} = (g^{a_1})^{\prod(r_j, s_j)\in S(m) r_i} * r_1^{s_1} = g^{a_1 \prod(r_j, s_j)\in S(m) r_j} * (g^{k_1})^{k_1^{(-1)}} * (m - a_1 \prod(r_j, s_j)\in S(m)^{r_j}) \pmod p, c \quad (12)$$

【０１９４】によりPiの署名を検査する。

【０１９５】このように、全ての正直なサーバーは、署名 S(m) を発行している間に不正が行われた無効なサーバーのセット F⊂B を決定できる。サーバーはそれらの索引i ∈{1,n} により順序付けられるから、正直なサーバーが検出された不正な者を取り除くためにセットBを更新し、そして新しい活動状態のサーバーのセットを試

$$B \leftarrow (B \setminus F) \cup \{\max(B) + 1, \ldots, \max(B) + m\} \quad (13)$$

を取る。

【０１９７】本発明は、全てのラウンドで最大kのサーバーが敵対者に乗っ取られて無効となりうるから、後に続く更新は（不正をする者が”１対１”で現われる場合には）k回よりも多くは実行されないと想定する。これはラウンド毎に最大kの余分な署名を発行するのに等しい。これは無視できるオーバーヘッドである。

【０１９８】上記からの更新プロトコルは、ラウンドt毎に、秘密共有$x_i^{(t)}$ の公開対応部$y_i^{(t)}$ の維持を保証

$$y_i^{(t+1)} \leftarrow y_i^{(t)} * (g^{f_1(i)} g^{f_2(i)} \ldots g^{f_n(i)}) \pmod p \quad (14)$$

【０２００】Z_pの対数を計算する”難しさ”を仮定すれば、

【数１２８】

$$g^{f_i(j)}$$

【０２０１】の同報通信は敵対者を支援しない。

【０２０２】まとめとして、本発明の構成に関して以下の事項を開示する。

（１） 順向性、ロバスト及び回復可能な分散しきい値秘密共有を有する公開キー暗号化の方法であって、キーを形成するために通信ネットワークによりリンクされたサーバーを初期化するステップと、終了を有する別個のラウンドで動作するように前記サーバーを同期させるステップと、前記通信ネットワークで同報通信されたメッセージから前記ラウンドの前記終了で更新されたキーを計算するステップと、コンプロマイズされたサーバーのセットを形成するために前記更新されたキーを検査するステップと、前記コンプロマイズされたサーバーのセットを回復するステップとを含む方法。

（２） 前記初期化するステップは、前記サーバーの各々の乱数を選択するステップと、前記乱数から前記サーバーの各々の秘密値を計算するステップと、前記秘密値から前記サーバーの各々の秘密キーを計算するステップと、前記秘密キーの公開対応部を前記通信ネットワークで同報通信するステップとを含む、上記(１)に記載の方法。

（３） 前記更新されたキーを計算するステップは、前記サーバーの各々の多項式を定義する乱数のセットを取出すステップと、前記多項式から導き出された、前記サーバーの各々の新しい秘密キーを取出すステップと、前記多項式から導き出されたメッセージを前記通信ネットワークで同報通信するステップとを含む、上記(１)に記載の方法。

（４） 前記更新されたキーを検査する前記ステップは、無効なサーバーのセットを形成するために前記通信ネットワークで同報通信された前記メッセージを分析するステップと、前記無効なサーバーのセットに対応する告発のセットを生成するステップと、前記告発のセットを前記通信ネットワークで同報通信するステップと、更新された無効なサーバーのセットを形成する前記同報通信された告発のセットを分析するステップとを含む、上記(１)に記載の方法。

（５） 前記コンプロマイズされたサーバーのセットを識別するメッセージをコンソールに表示するステップを更に含む、上記(１)に記載の方法。

（６） 前記サーバーのセットを回復するステップは、前記コンプロマイズされたサーバーのセットにある各サーバーに新しい秘密キーを設定するステップと、前記サーバーから回復サーバーのセットを選択するステップと、前記回復サーバーのセットにある各サーバーの下位共有すことは自明である。例えば、それらが

【数１２３】B = 1, 2,..., k + 2により開始する全てのラウンドの始めに、そしてそれらが署名して m>0 の不正な者のセット

【数１２４】$F = \{i_1, i_2, \ldots, i_m\}$

【０１９６】を検出する毎に、それらは

【数１２５】

$$B \leftarrow (B \setminus F) \cup \{\max(B) + 1, \ldots, \max(B) + m\} \quad (13)$$

するように拡張できる。ステップ(2) で、サーバーPiが $f_i(j)$ をPjに送るとき、それは

【数１２６】

$$g^{f_i(j)}$$

【０１９９】を全ての他のサーバーにも送る。これは全てのサーバーが次の式のように全てのｉの公開キーを計算することを可能にする。

【数１２７】

$$y_i^{(t+1)} \leftarrow y_i^{(t)} * (g^{f_1(i)} g^{f_2(i)} \ldots g^{f_n(i)}) \pmod p \quad (14)$$

法。

を計算するステップと、前記下位共有から取出されたメッセージを前記通信ネットワークで同報通信するステップと、前記コンプロマイズされたサーバーのセットが受取った前記下位共有から取出された前記メッセージを検査するステップとを含む、上記(１)に記載の方法。

(７) 回復サーバーのセットを選択する前記ステップは、前記サーバーの下位セットを選択するステップと、前記下位セットの各サーバーの乱数値のセットを計算するをステップと、前記乱数値のセットから取出された署名されたメッセージを前記通信ネットワークで同報通信するステップと、無効サーバーのセットを取出すために前記署名されたメッセージを検査するステップと、前記サーバーの前記下位セットから前記無効サーバーのセットを除去するステップとを含む、上記(６)に記載の方法。

(８) 前記サーバーの各々の乱数を取出すステップと、前記乱数から前記サーバー毎の第１の署名の部分を計算するステップと、前記第１の署名の部分を前記通信ネットワークで同報通信するステップと、前記第１の署名の部分から前記サーバーの各々の第２の署名の部分を計算するステップとを更に含む、上記(１)に記載の方法。

(９) 署名されたメッセージを前記第２の署名の部分により検査するステップを更に含む、上記(８)に記載の方法。

(１０) 順向性、ロバスト及び回復可能な分散しきい値秘密共有を有する公開キー暗号方式を処理するデータ処理システムであって、通信ネットワークによりリンクされたサーバーと、前記サーバーと関連したキーを形成するために前記サーバーを初期化する初期化手段と、終了を有する別個のラウンドに前記サーバーの動作を同期させるタイミング手段と、更新されたキーを生成するために前記別個のラウンドの各ラウンドの終了で前記キーを更新する更新手段と、コンプロマイズされたサーバーのセットを形成するために前記更新されたキーを検査する検査手段と、前記コンプロマイズされたサーバーのセットを回復する回復手段とを備えるデータ処理システム。

(１１) 前記サーバーの各々の乱数を選択する選択手段と、前記乱数から前記サーバーの各々の秘密値を計算する第１の計算手段と、前記秘密値から前記サーバーの各々の秘密キー及び前記秘密キーの公開対応部を計算する第２の計算手段と、前記秘密キーの各々の前記公開対応部を前記通信ネットワークで同報通信する同報通信手段とを備える、上記(１０)に記載のデータ処理システム。

(１２) 前記更新手段は、前記サーバーの各々に関連した、多項式を定義する乱数のセットを生成する乱数生成手段と、前記多項式から前記サーバーの各々の新しい秘密キーを生成する秘密キー生成手段と、前記通信ネットワークで前記多項式から取出されたメッセージを同報通信する同報通信手段とを備える、上記(１０)に記載のデータ処理システム。

(１３) 前記検査手段は、無効なサーバーのセットを形成するために前記通信ネットワークで同報通信されたメッセージを分析する第１の分析手段と、前記無効なサーバーのセットに対応する告発のセットを生成する告発手段と、前記告発のセットを前記通信ネットワークで同報通信する同報通信手段と、無効なサーバーの更新されたセットを形成するために前記同報通信された告発のセットを分析する第２の分析手段とを備える、上記(１０)に記載のデータ処理システム。

(１４) 前記コンプロマイズされたサーバーのセットを表示する表示手段を更に備える、上記(１０)に記載のデータ処理システム。

(１５) 前記回復手段は、前記サーバーから回復サーバーのセットを選択する選択手段と、前記回復サーバーのセットにあるサーバーの各々の下位共有を計算する計算手段と、前記回復サーバーのセットにあるサーバーの各々の前記下位共有から取出されたメッセージを前記通信ネットワークで同報通信する同報通信手段と、前記コンプロマイズされたサーバーのセットが受取った前記下位共有の各々から取出された前記メッセージを検査する検査手段とを更に備える、上記(１０)に記載のデータ処理システム。

(１６) 前記選択手段は、前記サーバーの下位セットを選択する選択手段と、前記下位セットにある各サーバーの乱数値のセットを計算する計算手段と、前記乱数値のセットから取出された署名されたメッセージを前記通信ネットワークで同報通信する同報通信手段と、無効なサーバーのセットを取出すために前記署名されたメッセージを検査する検査手段と、前記無効なサーバーのセットを前記下位セットから除去する除去手段とを備える、上記(１５)に記載のデータ処理システム。

(１７) 前記サーバーの各々の乱数のセットを生成する乱数生成手段と、前記乱数のセットから前記サーバーの各々の署名の第１の部分を計算する第１の計算手段と、前記署名の第１の部分を前記通信ネットワークで同報通信する同報通信手段と、前記第１の署名の部分から前記サーバーの各々の署名の第２の部分を計算する第２の計算手段とを更に備える、上記(１０)に記載のデータ処理システム。

(１８) 署名されたメッセージを前記第２の署名の部分により検査する検査手段を更に備える、上記(１７)に記載データ処理手段。

【図面の簡単な説明】

【図１】サーバーコンソールを示す図である。

【図２】通信チャンネルに接続するサーバーを示すブロック図である。

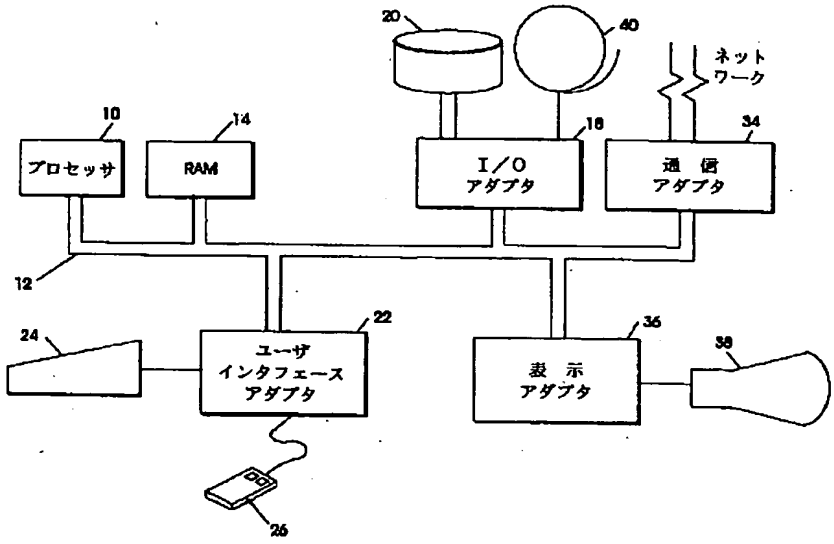
【図３】本発明の良好な実施例の高レベルのステップのシーケンスを示す流れ図である。

【図４】本発明の共有更新プロトコルの良好な実施例のステップのシーケンスを示す流れ図である。

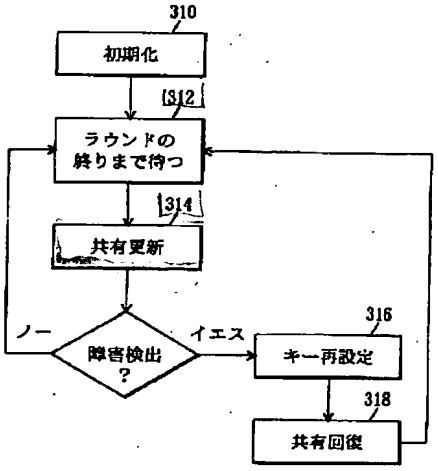
【図５】図４のステップ410 におけるステップのシーケンスを示す流れ図である。
【図６】本発明の共有回復プロトコルの良好な実施例のステップのシーケンスを示す流れ図である。
【図７】本発明の共有回復プロトコルの良好な実施例のステップのシーケンスを示す流れ図である。
【符号の説明】

10	中央処理装置
12	システムバス
14	ランダムアクセスメモリ(RAM)
18	入出力(I/O) アダプタ
20	ディスク装置
22	ユーザインタフェースアダプタ
24	キーボード
26	マウス
34	通信アダプタ
36	表示アダプタ
38	表示装置
40	テープ装置

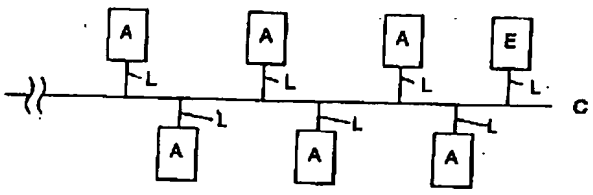
【図1】



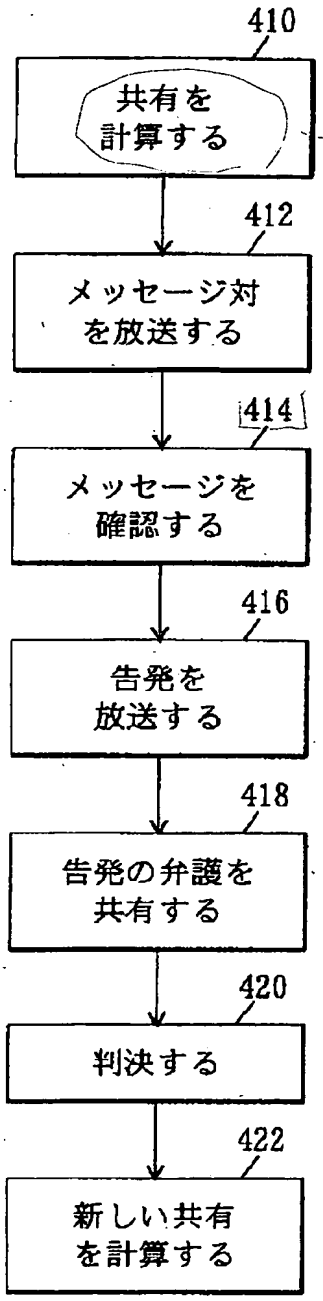
【図3】



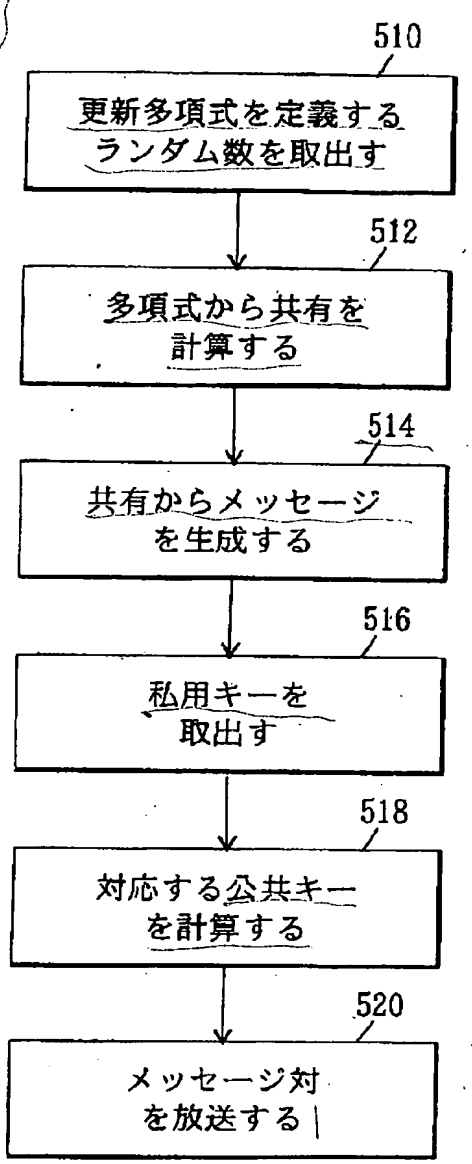
【図2】



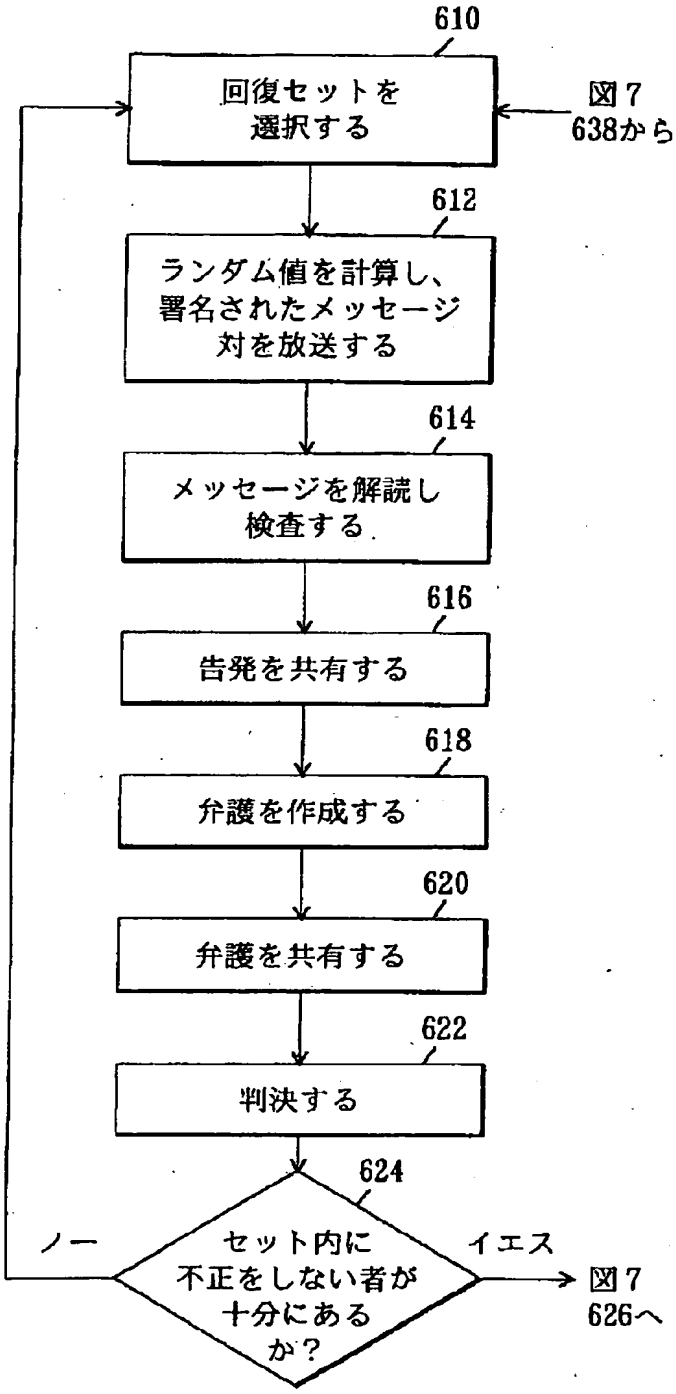
【図4】


410
共有を計算する
412
メッセージ対を放送する
414
メッセージを確認する
416
告発を放送する
418
告発の弁護を共有する
420
判決する
422
新しい共有を計算する

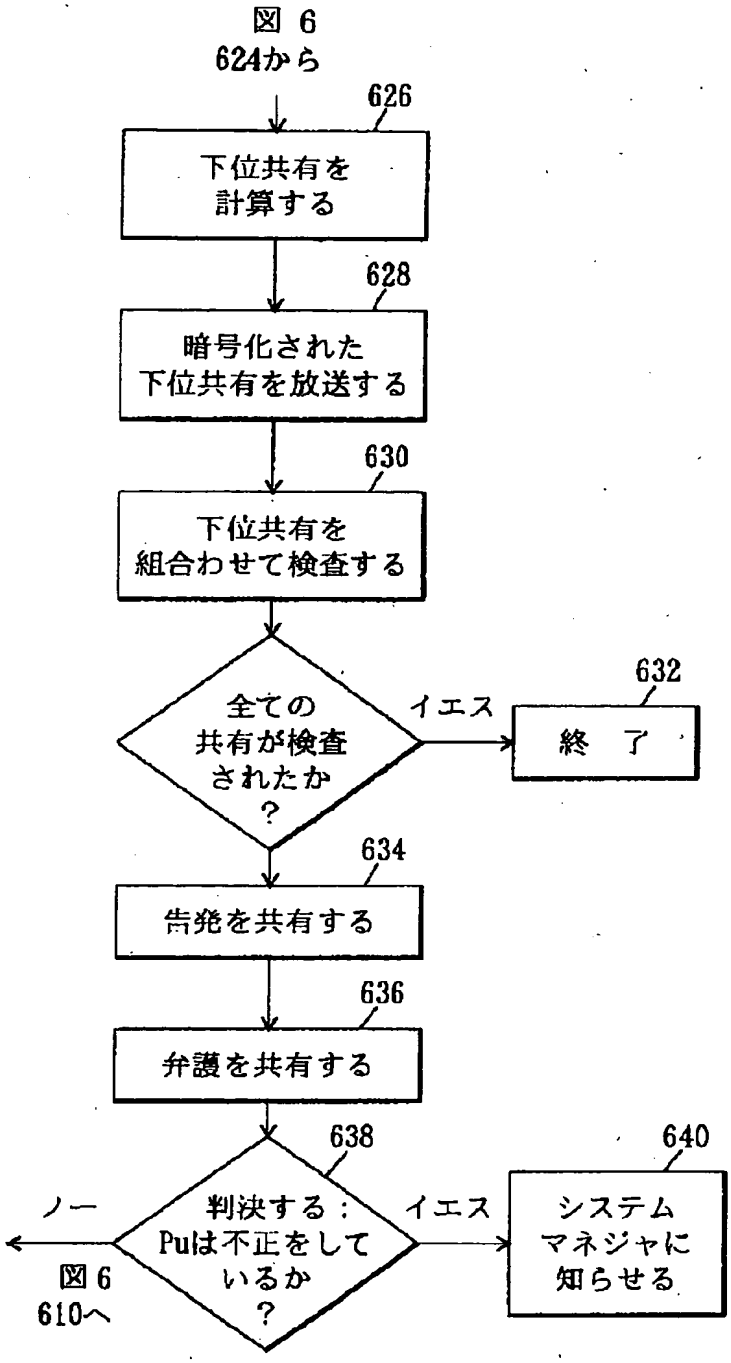
【図5】


510
更新多項式を定義するランダム数を取出す
512
多項式から共有を計算する
514
共有からメッセージを生成する
516
私用キーを取出す
518
対応する公共キーを計算する
520
メッセージ対を放送する

【図6】


610
回復セットを選択する
図7 638から
612
ランダム値を計算し、署名されたメッセージ対を放送する
614
メッセージを解読し検査する
616
告発を共有する
618
弁護を作成する
620
弁護を共有する
622
判決する
624
セット内に不正をしない者が十分にあるか？
ノー
イエス
図7 626へ

【図7】



フロントページの続き

(51)Int.Cl.6	識別記号	庁内整理番号	FI	技術表示箇所
			H04L 9/00	601F

(72)発明者 スタニスロー・マイケル・ヤレッキー
アメリカ合衆国マサチューセッツ州、ケンブリッジ、マサチューセッツ・アベニュー 282、アパートメント・423・ディ

(72)発明者 ヒューゴ・マリオ・クロウジック
アメリカ合衆国ニューヨーク州、リバーデイル、ネザーランド・アベニュー 2600

(72)発明者 マーセル・モードチャイ・ユング
アメリカ合衆国ニューヨーク州、ニューヨーク、ワシントン・ストリート 1